

PROCESSOR SURROGATE FOR USE IN MULTIPROCESSOR SYSTEMS
AND MULTIPROCESSOR SYSTEM USING SAME

TECHNICAL FIELD

[0001] The present invention generally relates to data processing systems, and more particularly relates to multiprocessor systems.

BACKGROUND

[0002] Over the history of digital computers there has been a continuous trend toward higher performance. Recent advances in integrated circuit (IC) manufacturing technology have produced smaller and faster ICs such that current microprocessor-based computer systems have higher performance than supercomputers of a generation ago. Microprocessor performance is determined by many factors, including clock speed and data bus width.

[0003] Typically IC manufacturers have been able to offer higher speed versions of a particular microprocessor over its lifetime. The continual improvement in microprocessor speed has created the opportunity for users to upgrade their computer systems with newer, higher speed microprocessors. Thus an older, slower microprocessor could be unplugged from its socket and a new, higher speed microprocessor inserted in its place. An example of this type of upgradeability is a microprocessor that can communicate with a memory device at one speed but whose internal clock speed can be scaled to a higher frequency, as disclosed in U.S. Patent No. 5,828,869 to Johnson et al.

[0004] This type of upgrading has allowed significant performance increases in single processor systems. However recent computer architectures have become more complex than single processor systems. For example some computer architectures now use multiple processors and non-uniform memory accessing (NUMA). In such NUMA systems, two or more microprocessors are connected in a ring or network and each have associated memory and possibly one or more associated input/output devices. It would be desirable for users to begin with a low-cost NUMA system and upgrade the system to improve performance later.

[0005] Accordingly, it would be desirable to provide a new means of upgrading performance in multiprocessor computer systems. This and other desirable features and characteristics of the present invention will become apparent from the subsequent detailed description and the appended claims, taken in conjunction with the accompanying drawings and the foregoing technical field and background.

BRIEF SUMMARY

[0006] A processor surrogate is adapted for use in a processing node of a multiprocessor data processing system having a plurality of processing nodes coupled together and to a plurality of input/output devices using corresponding communication links. The processor surrogate includes a first port comprising a first set of integrated circuit terminals adapted to be coupled to a first external communication link for coupling to one of the plurality of processing nodes, a second port comprising a second set of integrated circuit terminals adapted to be coupled to a second external communication link for coupling to one of the plurality of input/output devices, and an interconnection circuit coupled between the first port and the second port.

[0007] In another form a multiprocessor data processing system includes first and second processing nodes and an input/output device. The first processing node includes an actual processor. The second processing node includes a processor surrogate. The processor surrogate has a first port coupled to the first processing node, a second port, and an interconnection circuit coupled between the first port and the second port. The input/output device is coupled to the second port of the second processing node and is accessible to the actual processor via the processor surrogate.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The present invention will hereinafter be described in conjunction with the following drawing figures, wherein like numerals denote like elements, and

[0009] FIG. 1 illustrates a block diagram of a multiprocessor computer system useful in understanding the present invention;

[0010] FIG. 2 illustrates a block diagram of a portion of the multiprocessor computer system of FIG. 1 including one of the processors and its associated memory;

[0011] FIG. 3 illustrates a block diagram of a multiprocessor computer system using a processor surrogate according to the present invention;

[0012] FIG. 4 illustrates in a block diagram of the processor surrogate of FIG. 3;

[0013] FIG. 5 illustrates a block diagram of a multiprocessor computer system using a processor surrogate according to another aspect of the present invention;

[0014] FIG. 6 illustrates a block diagram of the processor surrogate of FIG. 5;

[0015] FIG. 7 illustrates a block diagram of a multiprocessor computer system using the processor surrogate of FIG. 6 according to yet another aspect of the present invention;

[0016] FIG. 8 illustrates a top view of an integrated circuit package that may be used for the actual processor of FIG. 2 and the processor surrogates of FIGs. 4 and 6;

[0017] FIG. 9 illustrates a side view of the integrated circuit package of FIG. 8; and

[0018] FIG. 10 illustrates a bottom view of the integrated circuit package of FIG. 8.

DETAILED DESCRIPTION

[0019] The following detailed description is merely exemplary in nature and is not intended to limit the invention or the application and uses of the invention. Furthermore, there is no intention to be bound by any expressed or implied theory presented in the preceding technical field, background, brief summary or the following detailed description.

[0020] FIG. 1 illustrates a block diagram of a multiprocessor computer system 100 useful in understanding the present invention. Computer system 100 includes two processor nodes represented by circles including a first processor node labeled "P0" and a second processor node labeled "P1" connected together over a communication link 116. Nodes P0 and P1 are implemented using microprocessors 110 and 120, respectively. System 100 also includes a first input/output (I/O) device 130 labeled "I/O A", a second I/O device 140 labeled "I/O B", a third I/O device 150 labeled "I/O C", a fourth I/O device 160 labeled "I/O D", a first

dynamic random access memory (DRAM) 170 labeled “DRAM 0”, and a second DRAM 180 labeled “DRAM 1”. Processor 110 is a single-chip microprocessor that communicates with I/O devices 130 and 140 over communication links 112 and 114, respectively, and with processor 120 over link 116. Processor 110 also has a dedicated bus for performing memory accesses with local DRAM 170. Likewise processor 120 communicates with I/O devices 150 and 160 over corresponding links and has a dedicated bus for connection to local DRAM 180. I/O devices 130, 140, 150, and 160 may be any of a variety of I/O devices including a graphics processor, an Ethernet controller, a bridge to another bus such as that specified by the Personal Computer Interconnect (PCI) Special Interest Group, etc.

[0021] Processors 110 and 120 communicate with their respective I/O devices using link controllers complying with the HyperTransport™ I/O Link Specification, Revision 1.05, © 2003 HyperTransport Technology Consortium, and are able to achieve a throughput of 3.2 GB/sec when using a 1600 MHz data rate. HyperTransport technology is a packet-based link implemented on two independent unidirectional sets of wires. So for example links 112, 114, and 116 include both an output connection and an input connection. Each HyperTransport link is nominally point-to-point and connects two devices. Chains of HyperTransport links can also be used as an I/O channel, connecting I/O devices and bridges to a host system. The HyperTransport link is designed to deliver a high-performance and scalable interconnect between CPU, memory, and I/O devices. The HyperTransport link uses low swing differential signaling with on-die differential termination to achieve very high data rates. The HyperTransport link uses scalable frequency and data width to achieve scalable bandwidth.

[0022] System 100 includes memory associated with each processor node and distributed among the nodes. It implements a cache coherent non-uniform memory access (“CC NUMA”) architecture. The CC NUMA architecture is non-uniform in that all memory in the system is visible to each processor, but access time depends on the physical distance between the processor and the memory. Thus processor 110 is able to access DRAM 170 quickly, but must wait while the memory access request travels over link 116 before it can access DRAM 180. Link 116 between processors 110 and 120 uses a special form of HyperTransport known as coherent HyperTransport.

[0023] FIG. 2 illustrates a block diagram of a portion 200 of multiprocessor computer system 100 of FIG. 1, including processor 110 and DRAM 170. Processor 110 is a single-

chip microprocessor and includes generally a central processing unit (CPU) 202, a memory controller 206, a crossbar switch labeled “XBAR” 208, and three link controllers 212, 214, and 216 each labeled “HT” for HyperTransport. CPU 202 is a processor adapted to execute instructions of the so-called x86 instruction set. The x86 instruction set is based on the instruction set of the 8086 microprocessor first manufactured by the Intel Corporation of Santa Clara, California. CPU 202 however includes many sophisticated functions for high-performance execution of x86 programs including pipelining and superscalar design. CPU 202 includes at least one cache 204 for storage of frequently used data. For example CPU 202 may include two top level (L1) caches, one for instructions and the other one for data, and a second level (L2) cache that is shared by the instruction and data streams.

[0024] Memory controller 206 is the mechanism for data transfer between processor 110 and DRAM 170. Memory controller 206 offloads the task of initiating and terminating memory accesses from CPU 202. It includes internal queues to allow efficient use of the external bus to DRAM 170. In other embodiments DRAM 170 could be replaced by a lower-level memory system including one or more additional caches and main memory, by static RAM, by nonvolatile memory, etc.

[0025] XBAR 208 is a switching/multiplexing circuit designed to couple together the buses internal to processor 110.

[0026] Link controllers 212, 214, and 216 are coupled to external links 112, 114, and 116, respectively. Links 112, 114, and 116 include output channels 220, 230, and 240, respectively, and input channels 222, 232, and 242, respectively. Each of link controllers 212, 214, and 216 comply with the HyperTransport™ I/O Link Specification, Revision 1.05, but additionally support the special coherent form of HyperTransport which is able to link two processor nodes.

[0027] Considering now FIGs. 1 and 2 together, it can be seen how processor 120 is able to access DRAM 170. Processor 120’s own memory controller, corresponding to memory controller 206, would receive a memory access request from its CPU. Upon recognizing that the access is for memory present at another node, it would send a memory access request to processor 110 through one of its link controllers over coherent link 116. The request packet would be received by link controller 216 and routed through XBAR 208 to memory controller 206. Memory controller 206 would then check its internal directory to

see whether the requested memory element is present in cache 204. If the requested memory element were absent from cache 204, memory controller 206 would read DRAM 170 and provide the requested data element through XBAR 208 and link controller 216 over coherent link 116 back to processor 120.

[0028] While it is possible to upgrade system 100 using socket-compatible but higher speed processors, more flexible upgrade capabilities would be desirable. Such a capability is shown in FIG. 3, which illustrates a block diagram of a multiprocessor computer system 300 using a processor surrogate 320 according to the present invention. As used herein “multiprocessor” means having more than one processing node, even though only one processing node may have an actual CPU. System 300 is similar to system 100 except that node P1 has been replaced by a node labeled “S1” having processor surrogate 320 that does not have its own CPU. As used herein, a “processor surrogate” is a device that plugs into the socket of node S1 and is used in place of an actual processor. By using processor surrogate 320 in place of an actual processor, an extra I/O device 350 is available for use in system 300, but without the expense of another actual microprocessor having its own CPU. System 300 is essentially a single-processor system that may be easily upgraded to a dual processor system. Thus system 300 is a lower cost system but has an upgrade path: an actual processor like that described with respect to FIG. 2 could later be inserted into the socket used by processor surrogate 320 to upgrade significantly the performance of computer system 300.

[0029] As will be seen hereinafter, a processor surrogate can be of two general types: active and passive. Both types of processor surrogates are socket compatible with and operate in place of an actual microprocessor but they differ in the type of interconnection circuit used. FIG. 4 illustrates a block diagram of processor surrogate 320 of FIG. 3. Processor surrogate 320 is a passive type surrogate and includes a first set of wires 390 to connect input signals of HyperTransport link 370 to output signals of HyperTransport link 380, and a second set of wires 392 to connect input signals of HyperTransport link 380 to output signals of HyperTransport link 370. Processor surrogate 320 includes integrated circuit terminals, corresponding to terminals of two link controllers of an actual microprocessor, such as microprocessor 110 of FIG. 2..

[0030] On power-up, processor 310 detects whether the HyperTransport link is coherent or non-coherent and negotiates the speed of information transfer over the link. Thus the link

controller in processor 310 connected to link 370 will detect, from communication with I/O controller 350 via processor surrogate 320, that the link is to be non-coherent. However if the processor surrogate were later replaced with an actual processor, then the link controller in processor 310 would detect the presence of an active node and configure it to be a coherent link.

[0031] Processor surrogate 320 has the same “footprint” as processor 110 of FIGs. 1 and 2, i.e. it is physically capable of plugging or being inserted into a socket that accommodates an actual processor like processor 110. Thus it would have the same integrated circuit package dimensions as an actual processor. However the integrated circuit package would only contain wires 390 and 392. In particular one type of package used for processor 110 is a package known as a ceramic micro pin grid array. To accommodate a socket for a micro pin grid array processor, processor surrogate 320 would also use a similar micro pin grid array package. However the pins for most signals would not be connected, and thus would exist as “dummy” pins. Pins 372, 374, 382, and 384 would be used to provide the appropriate interconnections. A ceramic type package provides the opportunity for forming interconnections using multiple signal planes within the ceramic material to reduce parasitic losses that would otherwise occur through the use of bond wires if a leadframe type package were used. Power and ground pins may optionally be connected to provide suitable ground planes for signal shielding from RF radiation and interference. Note that processor surrogate 320 may be redesigned to match the footprint of any other package type used for the actual processor. Also it would be possible to substitute a cheaper organic package for the ceramic package if the electrical and mechanical properties were sufficient.

[0032] More specifically, processor surrogate 320 may be housed in a ceramic micro pin grid array package having a total of 961 available pins formed as an array of 31 rows by 31 columns. Using HyperTransport links, input ports 372 and 384 each include thirty eight pins, including four clock input pins, two control input pins, and thirty two multiplexed command/address/data input pins, wherein each signal is conducted differentially using a pair of signal pins. Output ports 374 and 384 also include thirty eight corresponding pins, including four clock output pins, two control output pins, and thirty two multiplexed command/address/data output pins. To make processor surrogate 320, one would form internal interconnections 390 within the package of processor surrogate 320 by connecting the control input pins of the first link (link 0) and the control output pins of the second link (link 1), the clock input pins of link 0 and the corresponding clock output pins of link 1, and

the multiplexed command/address/data input pins of link 0 and the corresponding multiplexed command/address/data output pins of link 1. Similar connections 392 would be made for connecting the inputs of link 1 to the outputs of link 0. Note that a feature of HyperTransport is the ability to scale the number of command/address/data pins from two to thirty-two pairs of pins, and in other embodiments a different number of command/address/data pins besides the sixteen pairs of differential pins described above may be supported.

[0033] FIG. 5 illustrates a block diagram of a multiprocessor computer system 500 using a processor surrogate according to another aspect of the present invention. System 500 includes an actual processor 510 in node P0 and an active form of a processor surrogate 520 in node S1. Nodes P0 and S1 are connected together using a coherent HyperTransport link 590. System 500 includes four I/O devices, including an I/O device 530 labeled “I/O A”, an I/O device 540 labeled “I/O B”, an I/O device 550 labeled “I/O C”, and an I/O device 560 labeled “I/O D”. I/O devices 530 and 540 are connected to processor 510 using respective non-coherent HyperTransport links, and I/O devices 550 and 560 are connected to processor surrogate 520 using respective non-coherent HyperTransport links 592 and 594. System 500 also includes a first DRAM 570 labeled “DRAM 0”, and a second DRAM 580 labeled “DRAM 1”, connected respectively to nodes P0 and S1.

[0034] System 500 uses active processor surrogate 520 to allow more resources to be available to node P0 but without the need for a second actual processor having its own CPU and cache. As will be described further below processor surrogate 520 provides interconnection by active circuitry instead of simple wiring to allow processor 510 to access two I/O devices 550 and 560 and an extra DRAM 580 without needing an additional CPU. Since it lacks the CPU and cache, processor surrogate 520 is cheaper than an actual processor but provides an upgrade path for future performance enhancement.

[0035] The construction of an active processor surrogate is better understood with respect to FIG. 6, which illustrates a block diagram of a portion of system 500 of FIG. 5 including processor surrogate 520 and DRAM 580. As shown in FIG. 6, processor surrogate 520 includes a memory controller 606, a crossbar switch 608, and HyperTransport link controllers 612, 614, and 616 connected respectively to links 590, 592, and 594. As in processor 110 of FIG. 2, HyperTransport link controllers 612, 614, and 616 are connected to corresponding ports including sets of output connections 620, 630, and 640 and sets of input

connections 622, 632, and 634, respectively. HyperTransport link controllers 612, 614, and 616 are also connected to crossbar switch 608. Memory controller 606 is connected to crossbar switch 608 and to external DRAM 580.

[0036] The memory controllers, crossbar switches, and HyperTransport link controllers of both actual processor 110 of FIG. 2 and processor surrogate 520 are functionally identical. In the illustrated form crossbar switches 208 and 608 both include a feature to automatically detect the presence or absence of the CPU. Thus the design is modular and the integrated circuit used in processor surrogate 520 can be realized by simply deleting the CPU from the netlist and inputting the modified netlist to automated place and route CAD software. Since the CPU consumes a substantial majority of the integrated circuit area of processor 110, the integrated circuit used in processor surrogate 520 will be significantly less expensive. Alternatively the active processor surrogate could be formed using an actual processor with a defective CPU.

[0037] Note that for use with an active processor surrogate, link 590 uses the coherent form of HyperTransport. Like the memory controller, the link controllers in processor surrogate 520 are modular and are the same as those used in an actual processor. However on power up the link controller in processor surrogate 520 connected to processor 510 over link 590 detects an active device on the other end and configures the link into the coherent form of HyperTransport. Thus the protocol is appropriate for a surrogate with its own distributed memory and memory controller.

[0038] FIG. 7 illustrates a block diagram of a multiprocessor computer system 700 using the processor surrogate of FIG. 6 according to yet another aspect of the present invention. System 700 illustrates the flexibility of the active form of the processor surrogate in constructing complex system topologies having significant upgrade capabilities. System 700 includes four processing nodes labeled "P0", "S1", "S2", and "S3" implemented by an actual processor 710 and processor surrogates 720, 730, and 740, respectively. System 700 uses an actual processor such as processor 110 of FIG. 2 for P0, and active forms of processor surrogates such as processor surrogate 520 shown in FIG. 6 for nodes S1, S2, and S3. The processor nodes are connected in a ring using coherent HyperTransport links such that node P0 is connected to adjacent nodes S1 and S3, node S1 is connected to adjacent nodes P0 and S2, node S2 is connected to adjacent nodes S1 and S3, and node S3 is connected to adjacent nodes S2 and P0.

[0039] System 700 provides accessibility to DRAM and I/O devices connected to three active processor surrogates but without the need for additional CPUs required by actual processors. It also provides a significant upgrade path in the ability to expand the system to up to four processors.

[0040] In other systems other processor node topologies may be used and all such multiprocessor topologies may be populated with at least one actual processor and one or more processor surrogates to provide flexible upgrade paths. In addition note that while processor 110 of FIG. 2 supports communication over three HyperTransport links using three corresponding link controllers, in other embodiments an actual processor may include a different number of link controllers and the possibilities for use of processor surrogates vary as well. For example if actual processor 110 included four link controllers, a passive form of the processor surrogate could allow connection of processor 110 to two extra I/O devices connected to the processor surrogate in a two-node multiprocessor system. The availability of four link controllers also allows more complex network topologies than those specifically illustrated herein.

[0041] Note that the embodiments of the processor surrogate and multiprocessor systems were described herein in the context of a HyperTransport NUMA architecture. In other embodiments other inter-processor communication protocols may also be used. Furthermore note the inter-processor communication protocol need not be implemented using coherent links. For example software coherence management could be used with non-coherent inter-processor communication links. Moreover the disclosed microprocessor may implement other instruction sets than the x86 instruction set.

[0042] FIG. 8 illustrates a top view 800 of an integrated circuit package that may be used for actual processor 110 of FIG. 2 and for processor surrogate 320 of FIG. 4 and processor surrogate 520 of FIG. 6. The exemplary integrated circuit package is a micro pin grid array package. The pin grid array package is a package type that is especially well suited for replacement because it can be matched with a corresponding socket, and the processor surrogate can be easily extracted from it and replaced with an actual processor. From top view 800 it can be seen that the micro pin grid array package has a base portion 802 and a lid portion 804 within a central portion of the area defined by base portion 802. Base portion 802 has a bevel 806 in the upper right corner signifying the “A1” corner, as will be described more fully below.

[0043] FIG. 9 illustrates a side view 900 of the integrated circuit package of FIG. 8. From side view 900 is seen base portion 902 and lid 904. Beneath base portion 902 is a plurality of integrated circuit terminals in the form of an array of pins 906 extending downward from the bottom surface of base 902.

[0044] FIG. 10 illustrates a bottom view 1000 of the integrated circuit package of FIG. 8. From bottom view 1000 is seen A1 corner 1002 and an array of pins extending toward the viewer represented by solid circles. The package outlines a possible array of pins formed by columns 1004 and rows 1006. Rows 1004 include 31 rows and columns 1006 include 31 columns for a potential array of 961 pins. However, pins are not present in the corners and in several areas within the array so that the total number of pins is equal to 940. The rows are assigned letters in the sequence A, B, C . . . H, J . . . M, N, P, R, S . . . V, W, Y, AA, AB . . . AH, AJ, AK, AL from top to bottom and the columns are numbered from 1 to 31 from right to left.

[0045] Thus in one specific example, a normal processor has a pin assignment including the following pertinent pin assignments shown in TABLE I for link controller 212 and in TABLE II for link controller 214:

TABLE I

Pin Name	Location	Pin Name	Location	Pin Name	Location
L0_CADIN_L[0]	G2	L0_CADOUT_L[0]	AE1	L0_CLKIN_L[0]	L2
L0_CADIN_L[1]	H1	L0_CADOUT_L[1]	AE3	L0_CLKIN_L[1]	M5
L0_CADIN_L[2]	J2	L0_CADOUT_L[2]	AC1	L0_CLKIN_H[0]	L3
L0_CADIN_L[3]	K1	L0_CADOUT_L[3]	AC3	L0_CLKIN_H[1]	L5
L0_CADIN_L[4]	M1	L0_CADOUT_L[4]	AA3	L0_CLKOUT_L[0]	AA1
L0_CADIN_L[5]	N2	L0_CADOUT_L[5]	W1	L0_CLKOUT_L[1]	AB3
L0_CADIN_L[6]	P1	L0_CADOUT_L[6]	W3	L0_CLKOUT_H[0]	AB1
L0_CADIN_L[7]	R2	L0_CADOUT_L[7]	U1	L0_CLKOUT_H[1]	AB4
L0_CADIN_L[8]	H5	L0_CADOUT_L[8]	AF3	L0_CTLIN_L[0]	T1
L0_CADIN_L[9]	H4	L0_CADOUT_L[9]	AE5	L0_CTLIN_H[0]	R1
L0_CADIN_L[10]	K5	L0_CADOUT_L[10]	AD3	L0_CTLOUT_L[0]	U3
L0_CADIN_L[11]	K4	L0_CADOUT_L[11]	AC5	L0_CTLOUT_L[0]	U2
L0_CADIN_L[12]	M4	L0_CADOUT_L[12]	AA5		
L0_CADIN_L[13]	P5	L0_CADOUT_L[13]	Y3		
L0_CADIN_L[14]	P4	L0_CADOUT_L[14]	W5		
L0_CADIN_L[15]	T5	L0_CADOUT_L[15]	V3		
L0_CADIN_H[0]	G3	L0_CADOUT_H[0]	AF1		
L0_CADIN_H[1]	G1	L0_CADOUT_H[1]	AE2		
L0_CADIN_H[2]	J3	L0_CADOUT_H[2]	AD1		
L0_CADIN_H[3]	J1	L0_CADOUT_H[3]	AC2		
L0_CADIN_H[4]	L1	L0_CADOUT_H[4]	AA2		
L0_CADIN_H[5]	N3	L0_CADOUT_H[5]	Y1		
L0_CADIN_H[6]	N1	L0_CADOUT_H[6]	W2		
L0_CADIN_H[7]	R3	L0_CADOUT_H[7]	V1		
L0_CADIN_H[8]	G5	L0_CADOUT_H[8]	AF4		
L0_CADIN_H[9]	H3	L0_CADOUT_H[9]	AF5		
L0_CADIN_H[10]	J5	L0_CADOUT_H[10]	AD4		
L0_CADIN_H[11]	K3	L0_CADOUT_H[11]	AD5		
L0_CADIN_H[12]	M3	L0_CADOUT_H[12]	AB5		
L0_CADIN_H[13]	N5	L0_CADOUT_H[13]	Y4		
L0_CADIN_H[14]	P3	L0_CADOUT_H[14]	Y5		
L0_CADIN_H[15]	R5	L0_CADOUT_H[15]	V4		

TABLE II

Pin Name	Location	Pin Name	Location	Pin Name	Location
L1_CADIN_L[0]	B22	L1_CADOUT_L[0]	A4	L1_CLKIN_L[0]	B18
L1_CADIN_L[1]	A21	L1_CADOUT_L[1]	C4	L1_CLKIN_L[1]	E17
L1_CADIN_L[2]	B20	L1_CADOUT_L[2]	A6	L1_CLKIN_H[0]	C18
L1_CADIN_L[3]	A19	L1_CADOUT_L[3]	C6	L1_CLKIN_H[1]	E18
L1_CADIN_L[4]	A17	L1_CADOUT_L[4]	C8	L1_CLKOUT_L[0]	A8
L1_CADIN_L[5]	B16	L1_CADOUT_L[5]	A10	L1_CLKOUT_L[1]	C7
L1_CADIN_L[6]	A15	L1_CADOUT_L[6]	C10	L1_CLKOUT_H[0]	A7
L1_CADIN_L[7]	B14	L1_CADOUT_L[7]	A12	L1_CLKOUT_H[1]	D7
L1_CADIN_L[8]	E21	L1_CADOUT_L[8]	C3	L1_CTLIN_L[0]	A13
L1_CADIN_L[9]	D21	L1_CADOUT_L[9]	E4	L1_CTLIN_H[0]	A14
L1_CADIN_L[10]	E19	L1_CADOUT_L[10]	C5	L1_CTLOUT_L[0]	C12
L1_CADIN_L[11]	D19	L1_CADOUT_L[11]	E6	L1_CTLOUT_L[0]	B12
L1_CADIN_L[12]	D17	L1_CADOUT_L[12]	E8		
L1_CADIN_L[13]	E15	L1_CADOUT_L[13]	C9		
L1_CADIN_L[14]	D15	L1_CADOUT_L[14]	E10		
L1_CADIN_L[15]	E13	L1_CADOUT_L[15]	C11		
L1_CADIN_H[0]	C22	L1_CADOUT_H[0]	A3		
L1_CADIN_H[1]	A22	L1_CADOUT_H[1]	B4		
L1_CADIN_H[2]	C20	L1_CADOUT_H[2]	A5		
L1_CADIN_H[3]	A20	L1_CADOUT_H[3]	B6		
L1_CADIN_H[4]	A18	L1_CADOUT_H[4]	B8		
L1_CADIN_H[5]	C16	L1_CADOUT_H[5]	A9		
L1_CADIN_H[6]	A16	L1_CADOUT_H[6]	B10		
L1_CADIN_H[7]	C14	L1_CADOUT_H[7]	A11		
L1_CADIN_H[8]	E22	L1_CADOUT_H[8]	D3		
L1_CADIN_H[9]	C21	L1_CADOUT_H[9]	E3		
L1_CADIN_H[10]	E20	L1_CADOUT_H[10]	D5		
L1_CADIN_H[11]	C19	L1_CADOUT_H[11]	E5		
L1_CADIN_H[12]	C17	L1_CADOUT_H[12]	E7		
L1_CADIN_H[13]	E16	L1_CADOUT_H[13]	D9		
L1_CADIN_H[14]	C15	L1_CADOUT_H[14]	E9		
L1_CADIN_H[15]	E14	L1_CADOUT_H[15]	D11		

in which pin L0_CADIN_H[0] represents the high or more-positive pin of a differential pair of pins for control/address/data input pin 0 for link controller 0, L1_CLKOUT_L[1] represents the low or more-negative pin of the differential pair of pins for clock output pin 1 for link controller 1, and so on.

[0046] In order to make a passive form of the processor surrogate as shown in FIG. 4, one would need to internally connect the input terminals of link controller 0 to the corresponding output terminals of link controller 1, and the input terminals of link controller 1 to the corresponding output terminals of link controller 0. Thus, using the micro pin grid array example of FIGs. 8-10 and TABLEs I and II, pin L0_CADIN_L[0] (assigned to pin location G2) would be connected to pin L1_CADOUT_L[0] (assigned to pin location A4), pin L1_CADIN_H[15] (assigned to pin location E14) would be connected to pin L0_CADOUT_H[15] (assigned to pin location V4), and so on.

[0047] While at least one exemplary embodiment has been presented in the foregoing detailed description, it should be appreciated that a vast number of variations exist. It should also be appreciated that the exemplary embodiment or exemplary embodiments are only examples, and are not intended to limit the scope, applicability, or configuration of the invention in any way. Rather, the foregoing detailed description will provide those skilled in the art with a convenient road map for implementing the exemplary embodiment or exemplary embodiments. It should be understood that various changes can be made in the function and arrangement of elements without departing from the scope of the invention as set forth in the appended claims and the legal equivalents thereof.